

Tensor SimRank for Heterogeneous Information Networks

Ben Usman

Skolkovo Institute of Science and Technology,
Novaya St., 100, Skolkovo, 143025, Russian
Federation

Moscow Institute of Physics and Technology,
Institutskiy Lane 9, Dolgoprudny, Moscow,
141700, Russian Federation
ben.usman@skoltech.ru

Ivan Oseledets

Skolkovo Institute of Science and Technology,
Novaya St., 100, Skolkovo, 143025, Russian
Federation

Institute of Numerical Mathematics, Russian
Academy of Sciences, Gubkina St., 8, Moscow,
119333
i.oseledets@skoltech.ru

ABSTRACT

We propose a generalization of SimRank similarity measure for heterogeneous information networks. Given the information network, the intraclass similarity score $s(a, b)$ is high if the set of objects that are related with a and the set of objects that are related with b are pair-wise similar according to all imposed relations.

Categories and Subject Descriptors

[Information systems]: Retrieval models and ranking—*Similarity measures*

General Terms

SimRank, Probabilistic SVD, Tensor, Low-rank approximation

1. INTRODUCTION

Most data in the modern world can be treated as an information network, thus network node similarity measuring has wide range of applications: search [1], recommendation systems [2], research publication networks analysis [3], biology [4], transportation and logistics [5] and others.

Consider a semantic network: set of types \mathcal{T} , each type $t \in \mathcal{T}$ is a set of entities; set of relations \mathcal{R} , each relation is 2-order predicate defined on two types from \mathcal{T} :

$$\mathcal{R} \ni r_{tp} : t \times p \mapsto \{1, 0\}, t, p \in \mathcal{T},$$

both types in relation can be equal ($r_{tt} : t \times t \rightarrow \{0, 1\}$), few relations can share the same pair of types ($\exists r_{tp}^{(1)} \neq r_{tp}^{(2)} \in \{0, 1\}^{t \times p}$). That structure may be considered as a graph with colored vertices and colored edges: vertex color is its entity type, edge color corresponds to a relation.

The question that we address is how to define similarity functions

$$s_t : t \times t \rightarrow \mathbb{R}, \quad \forall t \in \mathcal{T},$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

21st ACM SIGKDD'15 Sydney

Copyright 2014 ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

A Type of: devised structured activity

Instance of: candidate KB completeness node, clarifying collection type, type of object, type of temporally stuff-like thing

Subtypes: board game, brand name game, card game, child's game, coin-operated game, dice game, electronic game, fantasy sports, game for two or more people, game of chance, guessing game, memory game, non-competitive game, non-team game, outdoor game, party game, puzzle game, role-playing game, sports game, table game, trivia game, word game

Instances: ducks and drakes, ultimate frisbee, darts, pachinko, Crossword Puzzle Activity CW, pool, Snooker, mini golf

Figure 1: OpenCyc ontology node of concept "Game"

that would reflect the closeness of objects based on "similarity of relations" they enter, and at the same time not mixing different relations as soon as "objects of different types and links carry different semantic meanings, and it does not make sense to mix them to measure the similarity without distinguishing their semantics" [6].

1.1 Related work

The basic graph structure similarity measure is the classical SimRank [7] over a homogeneous graph $G = (V, E)$ which is defined as follows:

$$N_G(a) = \{v \in V : (v, a) \in E(G)\},$$

$$s(a, b) = \frac{C}{I(a)I(b)} \sum_{\substack{v \in N(a) \\ w \in N(b)}} s(w, v).$$

The main drawback of this approach is that we cannot induce multiple relations or object types, so the only option is mixing them up into blobs "relation exists" and "all objects" that is completely not applicable in the case we have multiple relations with different semantics, for example the OpenCyc ontology node of the concept "Game" (see Figure 1) cannot be easily expressed via a single type of relations and objects.

Personalized PageRank [8] is also often used to measure

similarity in homogeneous graphs:

$$\pi_a(b) = \varepsilon \delta_a(b) + (1 - \varepsilon) \sum_{(w,b) \in E} \frac{\pi_a(w)}{\alpha_{w,v}},$$

that it same as PageRank, except random jumps are made into some pre-chosen node b , rather than into random node.

Another option is PathRank [6] that measures path-similarity between objects a, b picked from the same class A of the heterogeneous information network \mathcal{N} given a symmetric meta-path \mathcal{P} (set of paths that satisfy composition of relations M_1, M_2, \dots, M_n that $A \xrightarrow{M_1} C_1 \xrightarrow{M_2} C_2 \dots \xrightarrow{M_n} A$, so $A \xrightarrow{M_1 \circ M_2 \circ \dots \circ M_n} A$) as a number of paths from the object a to the object b (each step i must satisfy corresponding relation M_i in \mathcal{P}) normed over the number of paths from a to a plus the number of paths from b to b given \mathcal{P} :

$$s_{\mathcal{P}}(a, b) = \frac{\|\{p \in \mathcal{P} : a \xrightarrow{\mathcal{P}} b\}\|}{\|\{p \in \mathcal{P} : a \xrightarrow{\mathcal{P}} a\}\| + \|\{p \in \mathcal{P} : b \xrightarrow{\mathcal{P}} b\}\|}$$

That approach can handle several relations and object types and is very useful when we know the structure of relations we want our similarity measure to be based on. In case we want just to "put our relations into a black box" that would find similarity that would capture all network relations as a whole, we might want to use something different. Recently, an approach [9] for building an optimal linear combination of meta-paths has been proposed.

There are several works on measuring similarity between objects from different classes, see, for example, [10].

2. TENSOR SIMRANK

2.1 Problem statement

Let us consider a function $s_t(a, b)$ that assigns similarity score for two objects from the same class t as follows: objects $a, b \in t \in \mathcal{T}$ are similar (value $s_t(a, b)$ is high) if they relate to objects which are similar too. That interdependence can be expressed via the following definition:

$$N_{r_{tp}}(a) = \{b \in p \mid r_{tp}(a, b) = 1\},$$

$$s_t(a, b) = \frac{1}{Z} \sum_{r_{tp} \in \mathcal{R}} w(r_{tp}) \sum_{\substack{c \in N_p(a) \\ d \in N_p(b)}} s_p(c, d),$$

where r_{tp} is the relation between classes $t, p \in \mathcal{T}$, $N_{r_{tp}}$ is the neighbourhood function that returns set of objects from the class p that are related to the object a via the relation r_{tp} , $w(r_{tp})$ are the weights corresponding to the relation r_{tp} , Z is the normalization constant.

This can be rewritten as a *Tensor SimRank equation*:

$$s_{\alpha\beta} = \sum_{\gamma} w_{\alpha\beta\gamma} \mathbf{r}_{\alpha\beta\gamma} s_{\alpha\beta} \mathbf{r}_{\beta\alpha\gamma},$$

$$s = \text{diag}(\{s_t\}_{t \in \mathcal{T}}), \quad s_{\alpha\alpha} = 1,$$

where s is a block-diagonal matrix (one block per each entity type), w are the relation weights, $\mathbf{r}_{\alpha\beta\gamma}$ are the *stochastic relation tensors*¹ (which have non-zero blocks where relations exist).

¹We have to use tensors instead of matrices to have multiple relations on the same pair of classes

Similarity scores between elements of different classes are equal to zero by the definition. Relation between objects of unrelated classes is equal to zero by definition too. Equation (1) is basically the classical SimRank equation with the adjacency tensor instead of the adjacency matrix: each non-zero layer of tensor encodes some relation on the same pair of types. If one has more than a single relation between types $p, t \in \mathcal{T}$, then \mathbf{r} would have multiple non-zero layers on the intersection of indices associated with the classes t, p — one adjacency matrix per layer. In (1) the index γ stands for (weighted) summation over all layers of the tensor. That can be equivalently rewritten explicitly:

$$S = \sum_{\gamma} w_{\gamma} W_{\gamma} S W_{\gamma}^T + D, \quad (2)$$

where the diagonal matrix D has to be chosen in a such way that $\text{diag}(S) = I$.

2.2 Computational algorithm

Simple iterations for (1) are computationally demanding due to large-scale matrix-by-matrix products, thus we propose a method that exploits the fact that s is block diagonal and \mathbf{r} is a three-dimensional block tensor with size of the last dimension (number of layers) much less than the overall amount of objects. On each iteration k for each $r \in \mathcal{R}$ we recompute s_i updates independently (assuming all other s_j fixed), see Algorithm 1.

Algorithm 1: Idea under Tensor SimRank

Data: \mathcal{T} - classes, \mathcal{R} - relations

Result: $S = \{s_t(a, b)\}_{t \in \mathcal{T}}$

repeat

for $s_t \in S$ **do**

 assume all $S \setminus s_t$ fixed

for $r \in \mathcal{R} : r_{tp} : t \times p \mapsto \{1, 0\}$ **do**

for $(a, b) \in t$ **do**

for $(c, d) \in p$ **do**

$s_t^{next}(a, b) \leftarrow r_{tp}(a, c) s_p(c, d) r_{tp}(b, d)$
 $\quad \quad \quad \leftarrow r_{tp}(a, c) s_p(c, d) r_{pt}(d, b)$

end

end

end

end

 update all $s_t \leftarrow s_t^{next}$

until $\sum_t \|s_t - s_t^{next}\| < \delta$;

So we just update the similarity score for each class assuming all other classes similarities are fixed in a way that the objects from the target class (t) that are related to objects from some other class $(c, d) \in p$ that are close ($s_p(c, d)$ is high) become closer too ($s_t(a, b) \uparrow$).

To show actual vectorized algorithm of similarity computation, let us introduce some additional notations: set of entity types $\mathcal{T} = \{t_i\}_{i=0}^N$, each entity type t is a set of entities, set of symmetric relation functions $\mathcal{R} = \{r_{tp}^{(j)}\}_{j=0}^L$ where $r_{tp}^{(j)} : t \times p \rightarrow \{0, 1\}$, $t, p \in \mathcal{T}$, j is the order; column-stochastic matrix of pairwise types impacts (weights) $w \in \mathbb{R}^{N \times N}$; operator $W : r_{tp}^{(j)} \rightarrow \mathbb{R}^{\|t\| \times \|p\|}$ that maps relation

into corresponding column-stochastic adjacency matrix. If r_{tp} is not defined for some $(t, p) \in \mathcal{T}^2$, then $w_{tp} = 0$.

Algorithm 2: Vectorised Tensor SimRank for HSM

Data: \mathcal{T} - classes, \mathcal{R} - relations, w - relation weights
Result: $S = \{s_t(a, b)\}_{t \in \mathcal{T}}$
for $t \in \mathcal{T}$ **do**
 $s_t^{(0)} = I$
end
 $k = 0$
repeat
 for $t \in \mathcal{T}$ **do**
 $s_t^{new} = 0$
 for $\mathcal{R} \ni r : t \times p \mapsto \{1, 0\}$ **do**
 $s_t^{new} = s_t^{new} + w_{tp} W(r_{tp}) s_p^{(k)} W(r_{pt})$
 end
 $k = k + 1$
 end
 for $t \in \mathcal{T}$ **do**
 $s_t^{(k+1)} = s_t^{new} - \text{diag}(s_t^{new}) + I$
 end
until $\sum_{t \in \mathcal{T}} \|s_t^{(k+1)} - s_t^{(k)}\| \leq \varepsilon$;

To achieve better results (see above) on sparse relations we adopted the Low-Rank SimRank approximation [11] that uses Probabilistic Singular Value Decomposition [12] to perform fast approximate projections on low-rank matrix manifold at each step of the iterative process (Algorithm 3).

The only difference with Algorithm 2 is that on each step we perform probabilistic SVD decomposition of the matrix $S - I$, so that $S \approx I + UDU^T$, and project it onto the manifold of matrices of rank a_t .

2.3 Convergence conditions

Recall that the classical SimRank can be computed as a solution of the equation:

$$S := WSW^T - \text{diag}(WSW^T) + I.$$

Fixed-point iteration converges if W is a column-stochastic matrix. In the vector form ($\text{vec}(\cdot)$ operator maps an $n \times n$ matrix into a n^2 vector by taking column by column) that can be written as²:

$$[W \otimes W - I] \text{vec}(S) - \text{vec}(\text{diag}(WSW^T)) + \text{vec}(I) = 0,$$

if matrix W is stochastic, then $W \otimes W$ is stochastic too.

Tensor SimRank (2) computation can be equivalently written in the form:

$$S := \sum_{\gamma} w_{\gamma} W_{\gamma} S W_{\gamma}^T - \text{diag}(\sum_{\gamma} w_{\gamma} W_{\gamma} S W_{\gamma}^T) + I, \quad (3)$$

or in the vectorized for

$$[\sum_{\gamma} w_{\gamma} W_{\gamma} \otimes W_{\gamma} - I] \text{vec}(S) - \text{vec}(\text{diag}(\dots)) + \text{vec}(I) = 0.$$

Moreover, SimRank is also commonly approximated by the solution of the discrete Lyapunov equation:

$$S = cWSW^T + (1 - c)I,$$

² $\text{vec}(ABC) = (C^T \otimes A) \text{vec}(B)$

Algorithm 3: Low-rank Tensor SimRank for HSM

Data: \mathcal{T} - classes, \mathcal{R} - relations, w - relation weights, $\{a_t\}$ - approximation ranks
Result: $S = \{s_t(a, b)\}_{t \in \mathcal{T}}$
for $t \in \mathcal{T}$ **do**
 $s_t^{(0)} = I$
end
 $k = 0$
 $u_t = 0$
 $d_t = 0$
repeat
 for $t \in \mathcal{T}$ **do**
 $s_t^{new} = 0$
 for $\mathcal{R} \ni r : t \times p \mapsto \{1, 0\}$ **do**
 $s_t^{new} = s_t^{new} + w_{tp} (W(r_{tp}) W(r_{pt}) + W(r_{tp}) u_p d_p^T u_p^T W(r_{pt}))$
 end
 $k = k + 1$
 end
 for $t \in \mathcal{T}$ **do**
 $s_t^{new} = s_t^{new} - T$
 $u_t, d_t = \text{ProbabilisticSVD}(s_t^{new}, a_t)$
 $s_t^{(k+1)} = s_t^{new} + I$
 end
until $\sum_{t \in \mathcal{T}} \|s_t^{(k+1)} - s_t^{(k)}\| \leq \varepsilon$;

which can be generalized to the tensor case as

$$S = c \sum_{\gamma} w_{\gamma} W_{\gamma} S W_{\gamma}^T + (1 - c)I,$$

and a fixed-point iteration converges [13] if:

$$\sum_{\gamma=1} w_{\gamma} \|W_{\gamma}\|_1^2 \leq 1 \xleftrightarrow[\text{stochastic}]{\|W_{\gamma}\|_1=1} \sum_{\gamma} w_{\gamma} \leq 1.$$

We conjecture that fixed-point iterations for (3) converge if:

1. Each W_{γ} is stochastic
2. $\sum_{\gamma} w_{\gamma} = 1$

In the simplest form (we have no preferences among relations and classes) it reduces to (relations weight):

$$w_{tp} = \frac{1}{\sum_m \|\{r_{tm}^{(j)} \in \mathcal{R}\}\|}.$$

3. COMPUTATIONAL EXPERIMENT

3.1 Synthetic data: convergence test

To test convergence conditions we conducted series of tests on randomly generated sparse networks with different number of classes: $K \in \{3, 5, 7, 10\}$ and with randomly chosen number of objects in each $N_{real} \in U_{[N/2, N]}$, $N \in \{10 \dots 100\}$, full network of relation types (all possible types relations exists) with $2 \min(N_{real}^i, N_{real}^j)$ randomly chosen edges in each and default w matrix (no priority). All generated networks successfully converged that illustrates that convergent sufficient conditions listed in previous section were adequate, see Figures 2,3.

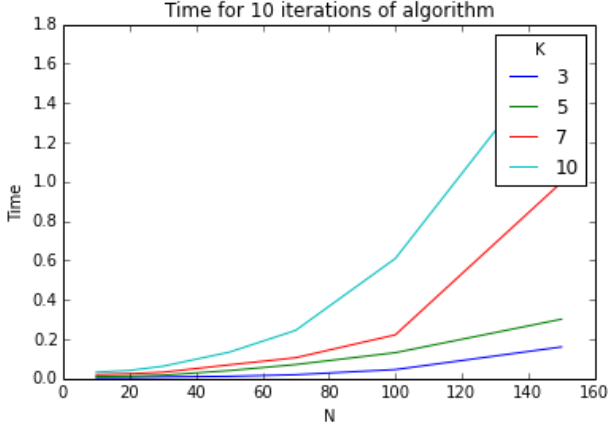


Figure 2: Average time spent on 10 iterations of algorithm on randomly network with K components, N objects in each

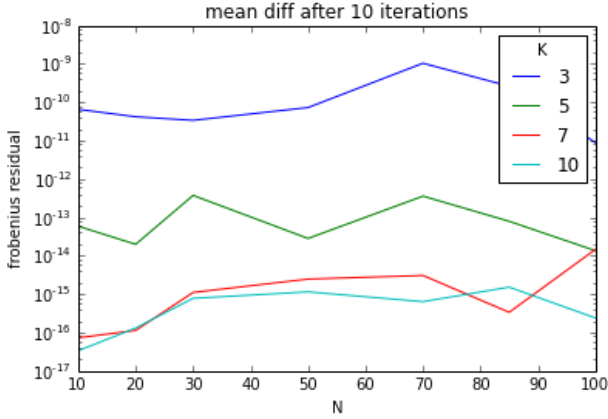


Figure 3: Mean Frobenius residual after 10 iterations of algorithm as function of number of objects (N), K components

3.2 Synthetic data: similarity reconstruction

To determine if model is capable of similarity reconstruction we generated a tree graph from randomly distributed points on a plane and tested if model can reconstruct points spatial similarity basing only on their relations.

On Figure 4 blue point represent 0-level point that are connected to 1-level point (red), that are connected to 2-level points (green).

We have measured the following similarity reconstruction \hat{S} quality compared to real S obtained from generated point coordinates:

$$Q(S, \hat{S}) = \frac{\sum_a \sum_b \sum_c [S_{ab} < S_{ac} \text{ and } \hat{S}_{ab} < \hat{S}_{ac}]}{\sum_i \sum_j \sum_k 1}$$

that actually shows how many "a is closer to c than to b" relations were preserved.

From Figure 5 one can see that at level $r \approx 0.3$ model gets saturated, but at the level $r \approx 0.15$ models that use low-rank version of Tensor SimRank perform way better than the "pure" algorithm. The numbers in the brackets

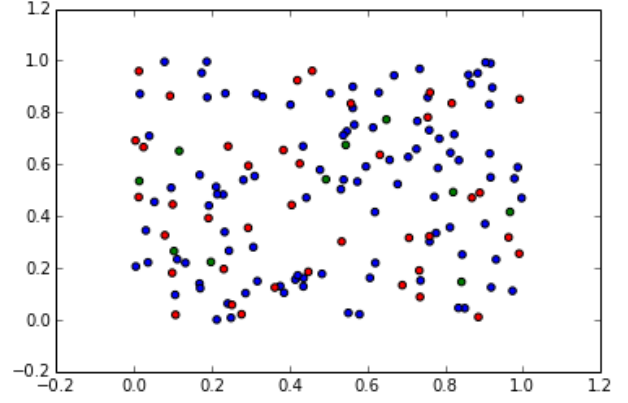


Figure 4: Random points for graph generation: blue points – zero level, red points – first level, green point – second level

Algorithm 4: Graph generation algorithm

Data: N - number of layers, $\{n_1, \dots, n_N\}$ - number of dots on each layer, r - connection radius

Result: \mathcal{T}, \mathcal{R}

for $k \in \{1..N\}$ **do**

$p^{(k)} \leftarrow$ generate n_k point from $U_{[0;1]}^2$

if $k > 0$ **then**

$\mathcal{R} \leftarrow r_k(p_i^{(k)}, p_j^{(k-1)})$ **if** $\rho(p_i^{(k)}, p_j^{(k-1)}) < r$

end

end

denote the dimensionality of the matrix space into which the similarity matrices were projected on each step (rank of approximation).

3.3 Book-Crossing Dataset test

The model was run on subsample from the Book-Crossing Dataset [14]. We have extracted only those authors who had highest (top100) number of books in the collection. The final network had the following structure:

$$\mathcal{T} = \{\text{Book, Author, Year, Publisher}\}$$

$$\mathcal{R} = \{\text{isAuthorOf}(\cdot, \cdot), \text{publishedBy}(\cdot, \cdot), \text{publishedIn}(\cdot, \cdot)\}$$

$$\#Book = 3625, \#Author = 99,$$

$$\#Year = 65, \#Publisher = 554$$

Model convergence is shown on Figure 3.3, where successful convergence to the best possible low-rank approximation can be seen. The similarity structure is clearly visible on Year similarity matrix heatmap (Figure (3.3)). We expect diagonal dominance as soon as temporarily close years should be more or less similar in terms of authors and publishers characteristic of that period. Tables 1 and 2 are examples of "closest book" requests, we want to notice that no NLP-preprocessing was conducted, nevertheless model treated books from same storybook as similar basing on author/publisher/year similarities.

4. DISCUSSION AND FURTHER WORK

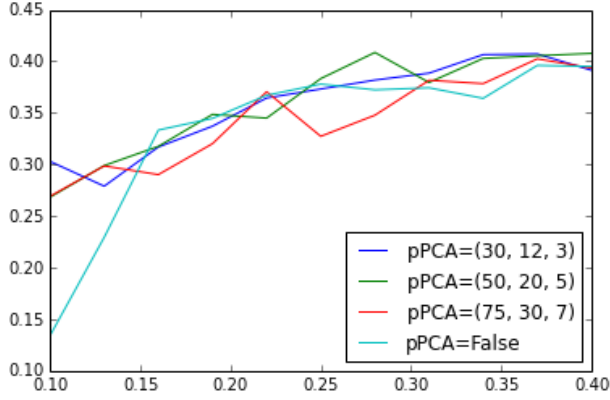


Figure 5: The value of $Q(S, \hat{S})$ as a function of r

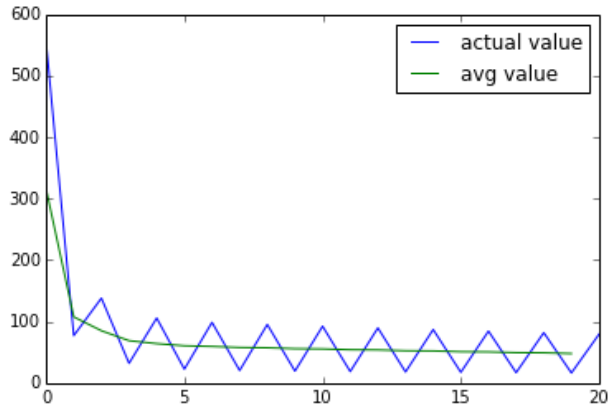


Figure 6: Monotonic reduction in the residual

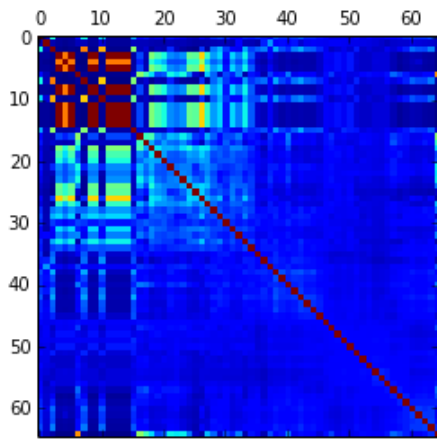


Figure 7: Year similarity matrix

Table 1: Books closest to "Psychic Sisters"

Psychic Sisters (Sweet Valley Twins and Friends, No 70)
The Love Potion (Sweet Valley Twins and Friends, No 72)
The Curse of the Ruby Necklace (Sweet Valley Twins and Friends Super, No 5)
She's Not What She Seems (Sweet Valley High No. 92)
Are We in Love? (Sweet Valley High, No 94)
Don't Go Home With John (Sweet Valley High No. 90)
In Love With a Prince (Sweet Valley High, No 91)

Table 2: Books closest to "The Girl Who Loved Tom Gordon"

The Girl Who Loved Tom Gordon
Hearts In Atlantis (All You Want to Know)
Blood And Smoke
Blood And Smoke Cd
Atlantis.
The Body (Penguin Readers: Level 5)
Storm of the Century

Proposed model can be used in various problem areas where most of the information is available in the form of relations between entities rather than features of individual entities and no trivial vector representation of those entities can be induced. One can use the vector representation

$$[s_t]_{ij} = \delta_{ij} + [u_t]_{ik}[d_t]_{kl}[u_t]_{lj},$$

to embed the notion of relations into classical machine learning algorithms. Also, the proposed model can be used for relation generalisation, that might give interesting results since we work on heterogeneous graphs.

Further model improvements might also include treating relations as objects too (probably, via heterogeneous hyper-graphs) and defining similarity matrix on relations.

5. CONCLUSION

This paper proposes the generalization of SimRank for heterogeneous networks and a method for its computation that exploits the fact that the resulting similarity matrix is block-diagonal, thus its components might be computed in an iterative fashion. The convergence conditions are proposed and successfully tested. Few perspective application areas are suggested.

6. REFERENCES

- [1] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web.," 1999.
- [2] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "GroupLens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

- [3] C. L. Giles, "The future of Citeseer: Citeseer X," in *Proceedings of the 10th European conference on Principle and Practice of Knowledge Discovery in Databases*, pp. 2–2, Springer-Verlag, 2006.
- [4] S. Roy, T. Lane, and M. Werner-Washburne, "Integrative construction and analysis of condition-specific biological networks.," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, p. 1898, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007.
- [5] W. Jiang, J. Vaidya, Z. Balaporia, C. Clifton, and B. Banich, "Knowledge discovery from transportation network data," in *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pp. 1061–1072, IEEE, 2005.
- [6] S. Lee, S. Park, M. Kahng, and S.-g. Lee, "Pathrank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems," *Expert Systems with Applications*, vol. 40, no. 2, pp. 684–697, 2013.
- [7] G. Jeh and J. Widom, "Simrank: a measure of structural-context similarity," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 538–543, ACM, 2002.
- [8] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th international conference on World Wide Web*, pp. 271–279, ACM, 2003.
- [9] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.
- [10] C. Shi, X. Kong, Y. Huang, S. Y. Philip, and B. Wu, "Hetesim: A general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge & Data Engineering*, no. 10, pp. 2479–2492, 2014.
- [11] I. V. Oseledets and G. V. Ovchinnikov, "Fast, memory efficient low-rank approximation of simrank," *CoRR*, vol. abs/1410.0717, 2014.
- [12] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.
- [13] J. Bierkens, O. v. Gaans, and S. V. Lunel, "Estimate on the pathwise lyapunov exponent of linear stochastic differential equations with constant coefficients," *Stochastic Analysis and Applications*, vol. 28, no. 5, pp. 747–762, 2010.
- [14] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, pp. 22–32, ACM, 2005.